

California State University, San Bernardino

CSUSB ScholarWorks

Theses Digitization Project

John M. Pfau Library

2004

Web based internship management system: A collaborative coordinating tool

Vishal Dharod

Follow this and additional works at: <https://scholarworks.lib.csusb.edu/etd-project>



Part of the [Software Engineering Commons](#)

Recommended Citation

Dharod, Vishal, "Web based internship management system: A collaborative coordinating tool" (2004).
Theses Digitization Project. 2575.
<https://scholarworks.lib.csusb.edu/etd-project/2575>

This Project is brought to you for free and open access by the John M. Pfau Library at CSUSB ScholarWorks. It has been accepted for inclusion in Theses Digitization Project by an authorized administrator of CSUSB ScholarWorks. For more information, please contact scholarworks@csusb.edu.

WEB BASED INTERNSHIP MANAGEMENT SYSTEM,
A COLLABORATIVE COORDINATING TOOL

A Project
Presented to the
Faculty of
California State University,
San Bernardino

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
in
Computer Science

by
Vishal Dharod
June 2004

WEB BASED INTERNSHIP MANAGEMENT SYSTEM,
A COLLABORATIVE COORDINATING TOOL

A Project
Presented to the
Faculty of
California State University,
San Bernardino


by
Vishal Dharod
June 2004

Approved by:




Dr. Ernesto Gomez, Chair, Computer Science

6/4/04
Date



Dr. Josephine Mendoza



Dr. David Turner

ABSTRACT

Since the dawn of the Internet, the computer industry has grown rapidly and has made a remarkable impact in the lives of everyone in a very short period of time. Major corporations, educational institutions, finance companies et al rely on several tools that run on a computer and save a lot of time and money. One such tool is presented in this project with the vision of being adapted by every educational institution to further ease their routine tasks. WICS (Web Based Internship Coordinating System) is a comprehensive toolkit for managing and coordinating internship programs in an educational institution. WICS was developed with the intension of providing an interactive tool for the faculty and students to communicate with each other whenever and wherever they want to. Students can undertand the requirements and view their progress and interact with the faculty in a better way. An instructor can review and update a student's progress and send email to students directly if necessary from WICS. Designed in PHP (Hypertext Preprocessor), an open source, server-side, HTML (Hypertext Markup Language) embedded scripting language used for creating dynamic web pages, this piece of software, is easy to maintain and further enhancements can be added without any hassles.

ACKNOWLEDGMENTS

I thank the faculty of Computer Science department for giving me an opportunity to pursue my Masters in Computer Science at the California State University, San Bernardino. I express my sincere appreciation to my graduate advisor, Dr. Ernesto Gomez who offered me this project and directed me through this entire effort. I also thank my other committee members, Dr. Josephine Mendoza and Dr. David Turner for their valuable support.

Also the support of National Science Foundation under award 9810708 is gratefully acknowledged.

TABLE OF CONTENTS

ABSTRACT	iii
ACKNOWLEDGMENTS	iv
LIST OF TABLES	viii
LIST OF FIGURES	ix
CHAPTER ONE: INTRODUCTION	
1.1 Introduction	1
1.2 Purpose of this Project	2
1.3 Project Products	3
CHAPTER TWO: ARCHITECTURE	
2.1 Project Design	4
2.2 Software Interfaces	6
CHAPTER THREE: DATABASE DESIGN	
3.1 Data Analysis	7
3.2 Database Schema Conceptual Model - Entity Relationship Diagram	8
3.3 Database Schema Logical Model - Relational Schema	9
3.4 Data Type and Details	9
CHAPTER FOUR: PROJECT IMPLEMENTATION	
4.1 Introduction	13
4.2 Graphical User Interface Design	14
4.1.1 Home Page	14
4.1.2 Login Page	15
4.1.3 Administrator Login Page	16
4.1.4 Administrator Add Faculty Page	17

4.1.5 Backup Database Page	18
4.1.6 Faculty Search Result Page	19
4.1.7 Faculty Login Page	20
4.1.8 Faculty View New Student Page	21
4.1.9 Faculty View Student Progress Page	22
4.1.10 Student Search Result Page	23
4.1.11 Company Search Result Page	24
4.1.12 Faculty Change Preferences Page	25
4.1.13 Error Page	26
4.1.14 Student Login Page	27
4.1.15 Student Company Search Page	28
4.1.16 Student View Progress Page	29
4.1.17 Submit Proposal Page	30
4.1.18 Proposal Submitted Page	31
4.1.19 Student Error Page	32
4.1.20 Internship Guidelines Page	33
4.1.21 Help Guide	34
CHAPTER FIVE: SYSTEM VALIDATION	
5.1 Unit Test	36
5.2 Subsystem Testing	40
5.3 System Testing	42
CHAPTER SIX: MAINTENANCE MANUAL	
6.1 Software Installation	43
6.1.1 RedHat 9.0 Installation	43

6.1.2 Database Installation	44
6.1.3 Hypertext Preprocessor Installation	47
6.1.4 Apache Web Server	47
6.1.5 Open Database Connectivity	48
6.2 Variables Modification	48
6.2.1 System Variables	49
6.3 System Installation/Migration	49
6.4 Backup and Restore	49
6.4.1 System Backup	50
6.4.2 Database Backup	50
6.4.3 System Restore	51
6.4.4 Database Restore	51
CHAPTER SEVEN: CONCLUSION AND FUTURE DIRECTIONS	
7.1 Conclusion	52
7.2 Future Directions	53
APPENDIX A: USER CLASS PRINTOUT	55
APPENDIX B: DATABASE CLASS PRINTOUT	57
REFERENCES	59

LIST OF TABLES

Table 1.	Structure of Table Users	10
Table 2.	Structure of Table Student	10
Table 3.	Structure of Table Company	11
Table 4.	Structure of Table Document	11
Table 5.	Structure of Table Submission	11
Table 6.	Structure of Table tempStudent	12
Table 7.	Unit Test Results (Forms)	37
Table 8.	Unit Test Results (Class: DataBase)	39
Table 9.	Subsystem Test Results	41
Table 10.	System Test Results	42

LIST OF FIGURES

Figure 1.	Architecture Diagram	4
Figure 2.	Enhanced Entity Relationship Diagram	8
Figure 3.	Entities and Attributes	9
Figure 4.	Use Case Diagram	13
Figure 5.	Home Page	15
Figure 6.	Login Page	16
Figure 7.	Administrator Login Page	17
Figure 8.	Add Faculty Page	18
Figure 9.	Backup Database Page	19
Figure 10.	Faculty Search Result Page	20
Figure 11.	Faculty Login Page	21
Figure 12.	View New Student Page	22
Figure 13.	View Student Progress Page	23
Figure 14.	Student Search Result Page	24
Figure 15.	Company Search Result Page	25
Figure 16.	Change Preferences Page	26
Figure 17.	Error Page	27
Figure 18.	Student Login Page	28
Figure 19.	Student Company Search Result Page	29
Figure 20.	Student View Progress Page	30
Figure 21.	Submit Proposal Page	31
Figure 22.	Proposal Submitted Page	32
Figure 23.	Student Error Page	33
Figure 24.	Internship Guidelines Page	34
Figure 25.	Help Guide	35

CHAPTER ONE

INTRODUCTION

1.1 Introduction

Web Based Internship Coordinating System (WICS) is a comprehensive tool for coordinating an internship class with the goal of facilitating continuous communication between the instructor and the student. By using WICS, students save a lot of time as they can communicate electronically with the instructor. A student wishing to take an internship class can access the system for getting information on companies having internship programs. The system also allows students to electronically submit a proposal to the instructor of their choice. An instructor has the ability to add and modify company information as well as view all the students they are advising for the internship. Instructors are also notified when a new student submits a proposal and the instructor has a choice of accepting or rejecting it. The instructor can update the progress of a student and students after successfully signing into the system, can view their progress and email the instructor should any questions arise. The administrator would be a caretaker of the entire system responsible for important tasks of backing up the database

and restoring an archived database if needed.

Administrator is the only one who can add faculty members to the system. All users of the system can update their personal information like password, email address and phone number.

1.2 Purpose of this Project

The purpose of this project is to design, build and implement an internship coordinating system with anytime and anywhere access availability. All user and company information will be stored in a Mysql database and retrieved by PHP and ODBC. The main purpose of this project is to provide an easy-to-use interface for students and faculty members to interact with each other during the course of a student's internship class. Students can view their progress in real time and get the latest information every time they access the system. Moreover, the system offers the authorization function to make sure that students can access his/her records only and all the other information is kept discrete. The student's progress page gives read-only information to the student about his/her progress whereas the instructor can update the student's progress page and email the student for any information if needed. In the system, all the

users can manage their own account information such as changing passwords or e-mail addresses. And furthermore, the users have the same login as their department login in order to help users remember their login information. The system assigns a unique login to every user by following the same convention used by the department.

1.3 Project Products

This project would lead to the following products:

- Implementation of WICS: a working web site with PHP scripts and Mysql database, which would achieve the needs of a communication tool for students registering for an internship class. All the pages containing an individual's information will be secure and will require authentication.
- Users manual: an implementation manual will be available for the user. Every page on the website has a link to a HELP guide. This HELP guide will answer all the frequently asked questions and also highlight ways of using the system.
- Systems Manual: a project report (this report) will be available with design details and specifications.

CHAPTER TWO

ARCHITECTURE

2.1 Project Design

This project, Web Based Internship Coordinating System (WICS), implements a web system to provide an environment for the students and the instructor to enable a student to successfully complete an internship class. Thus, the components needed to implement WICS are a database server, a web server, graphical user interface components, and a database interface Application Programming Interface (API) to programmatically access the database. Figure I describes the interaction among the components used in WICS.

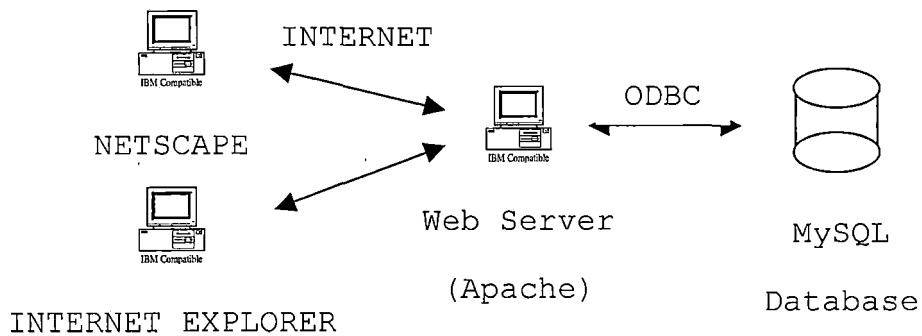


Figure 1. Architecture Diagram

The components used to build WICS were chosen with the following criteria: (1) the components should be shareware, i.e., available freely for non-commercial

purposes, (2) be part of a standard, i.e., they do not depend on a specific operating system and hence are easily portable across systems with ease, and (3) an independent database server, so that new and different versions of the server can be plugged in easily.

The user interface components are built by using HTML 6.0 forms, frames and Javascript. And the applications are launched using the HTML embedded scripting language PHP. PHP was used because it can be used in an object-oriented manner, which provides a reusable way for all programs and a web container called Apache that can be installed under Windows or Linux. Also, it is easy to process entire user input from the HTML forms. Moreover, PHP was used as it has the advantages of portability and efficiency. PHP is written just like any other object-oriented programming language and hence it is easy to maintain. PHP provides a convenient function, Open Database Connectivity (ODBC), to connect to the database.

The database used for WICS is Mysql. Mysql is a real multi-user database and is royalty-free open source software. To use it, simply activate Mysql in Linux. Also, the availability of the ODBC driver for Mysql is another important reason to choose it. Moreover, the same code could be used to link with another database by changing to

the proper ODBC driver, thereby making it database-independent.

2.2 Software Interfaces

- Internet browser: Netscape 4.7 or higher or Internet Explorer. 6.0 or higher
- Operating system: Windows 98/Me/2000/XP or Unix/Linux. (Red-Hat 7.2 or higher)
- Database: Mysql 4.0 or higher
- Language: HTML / JavaScript / PHP 4.7 or higher
- Database connector: ODBC.
- Web server: Apache 1.3 or higher

CHAPTER THREE

DATABASE DESIGN

3.1 Data Analysis

The data for designing and implementing the schema of the database depends on the properties of pages and users. The page data needed by the WICS pages are title, header, body and several tabs to move around the web pages. The user data needed by the system are the user social security number or student id, user name, password, first name, last name and e-mail address. The role of a user is automatically determined by the system at the time of login authentication and the user is automatically redirected to the appropriate page. All the user data will be checked on the server side when the user is created. All information entered by a user will be checked and appropriate error messages will be flagged by the System. Some of the important error messages flagged include password mismatch and missing required fields. A unique number identifies every record stored in the database and that serves as a parameter to check integrity constraints and maintain true data in the database.

3.2 Database Schema Conceptual Model - Entity Relationship Diagram

In designing the schema for the WICS database, two distinct parts have been identified. The first includes entities having relationship between them. The second includes a description of the entities and their attributes.

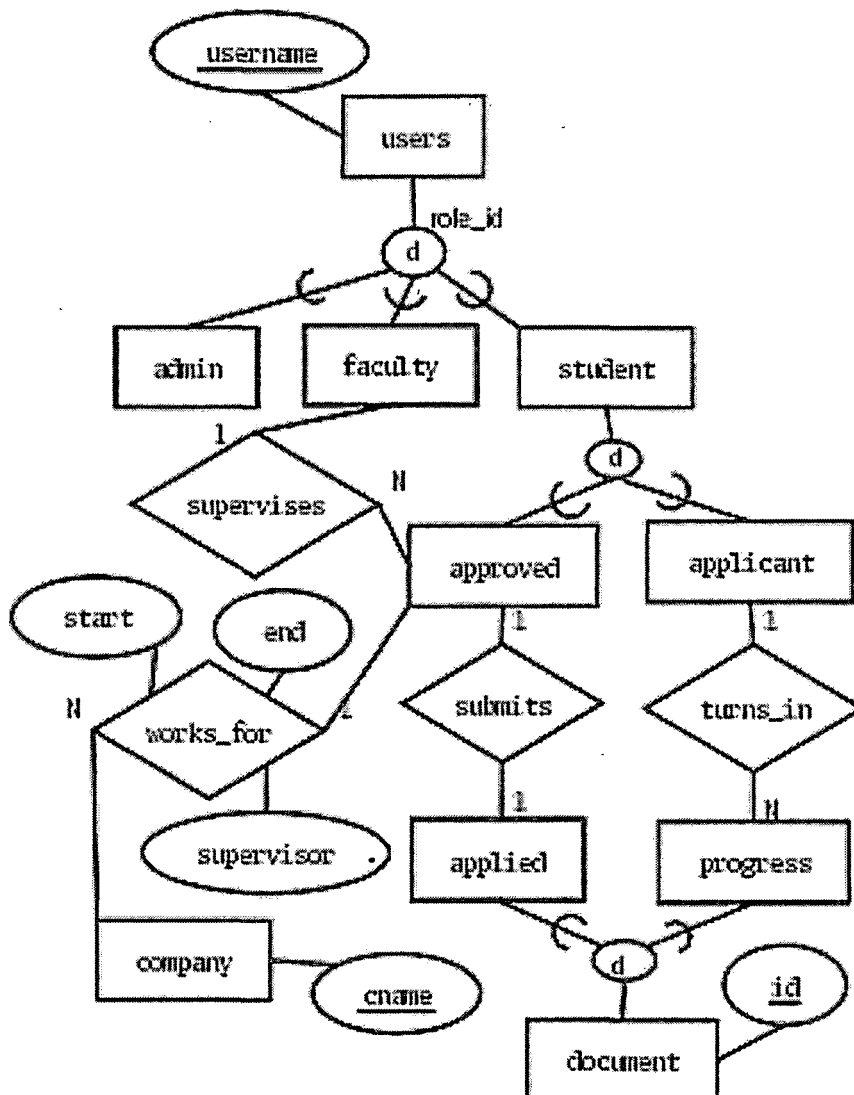


Figure 2. Enhanced Entity Relationship Diagram

3.3 Database Schema Logical Model - Relational Schema

The conceptual model EER diagram maps into the following relational table design. In the following tables, fields with keys indicate the primary key.

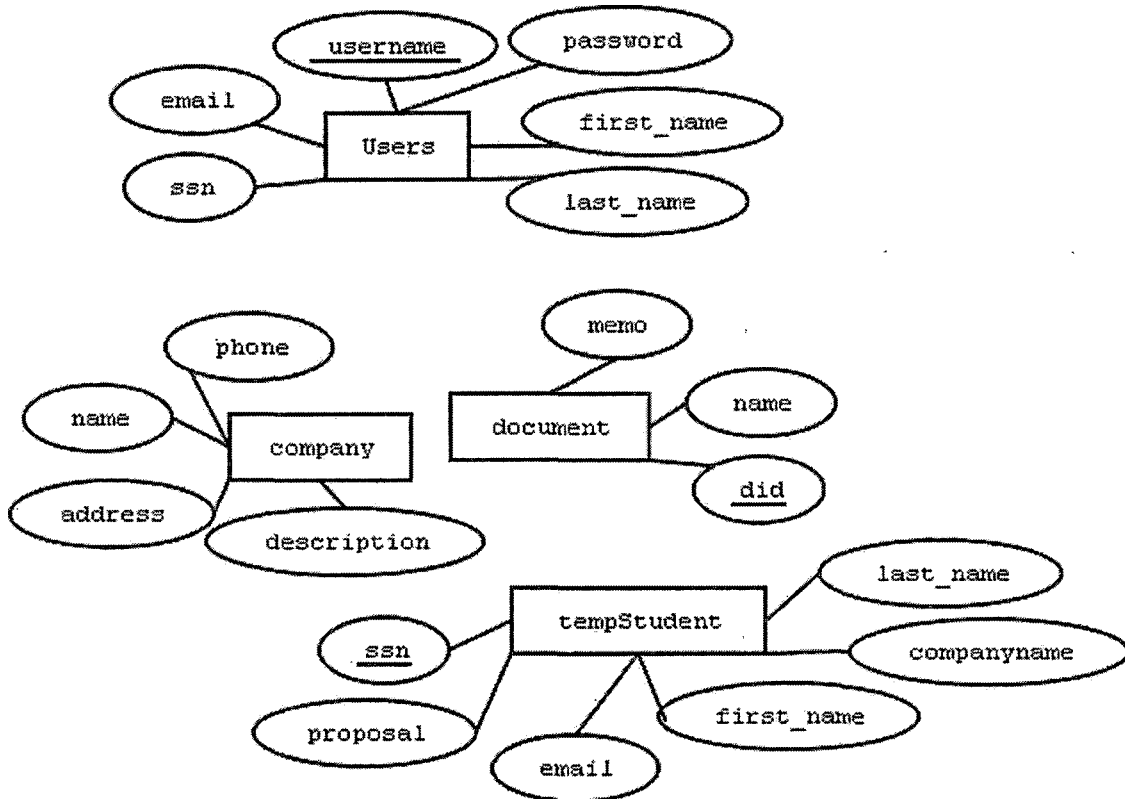


Figure 3. Entities and Attributes

3.4 Data Type and Details

The logical model establishes the following detailed design in Mysql database. The following tables describe data type, length, primary key, null or non-null keys, and extra information, such as auto_increment. The password field in the database is encrypted and it is shown in the

web pages as some special character repeated as many times as the length of the actual password. Fields like company description and document description are of the type memo, which will allow unlimited characters in a single row.

Table 1. Structure of Table Users

field	type	null	key	default	extra
username	varchar(15)		PRI		
password	varchar(32)	NO			
first_name	varchar(25)	NO			
middle_name	varchar(25)				
last_name	varchar(25)	NO			
role_id	int(3)				
ssn	int(10)				
email	varchar(60)				

Table 2. Structure of Table Student

field	Type	null	key	default	extra
ssn	int(10)	NO	PRI		
facultyssn	int(10)	NO			
Phone	int(10)	YES			
start_date	varchar(10)	NO			
end_date	varchar(10)	NO			
companyname	varchar(30)	NO	R		
Supervisor	varchar(20)	NO			

Table 3. Structure of Table Company

field	Type	null	key	default	extra
name	varchar(40)		PRI		
address	varchar(100)	NO			
phone	varchar(10)				
description	varchar(80)				

Table 4. Structure of Table Document

field	Type	null	key	default	extra
did	int(2)				
name	varchar(30)				
memo	varchar(40)				

Table 5. Structure of Table Submission

field	type	null	key	default	extra
studentid	int(10)		R		
documentid	int(2)		PRI		
date_submitted	varchar(10)				

Table 6. Structure of Table tempStudent

field	type	null	key	default	extra
companyname	varchar(30)	NO			
first_name	varchar(25)	NO			
proposal	varchar(100)				
last_name	varchar(25)	NO			
facultyssn	int(10)		R		
ssn	int(10)		PRI		
email	varchar(60)				

CHAPTER FOUR

PROJECT IMPLEMENTATION

4.1 Introduction

WICS is designed to perform several different functions for 3 different users.

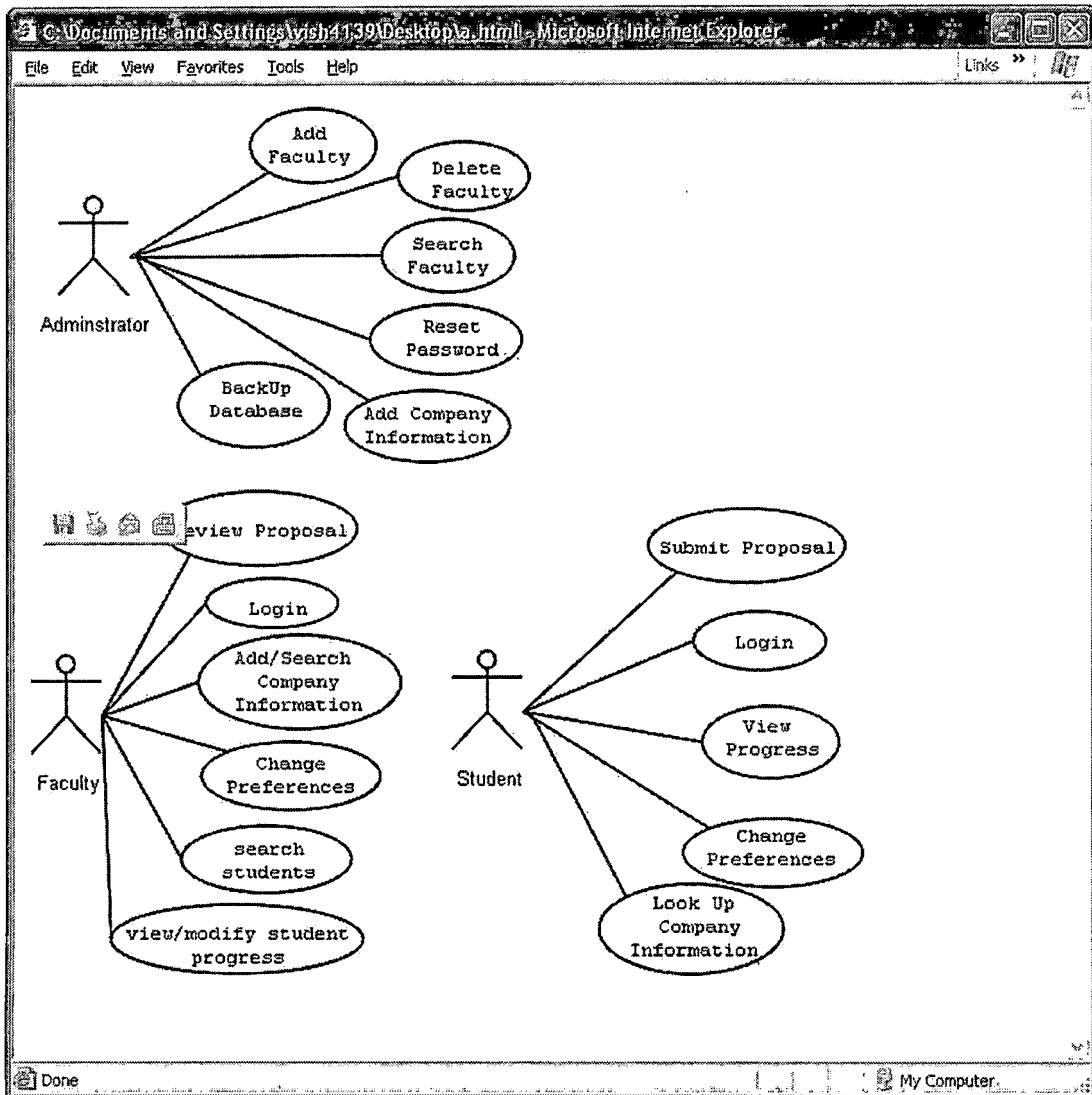


Figure 4. Use Case Diagram

4.2 Graphical User Interface Design

WICS GUI is easy to use. The GUI is written using Hyper Text Markup Language (HTML). All the functions that the user has are placed in the menu part, which is the left region of the page. It also uses JavaScript to check the user input accuracy. An error page will report all the inputs that are not acceptable by the system. Hence, the WICS GUI is executable with browsers that support javascript. The following sub sections explain the GUI work and details.

4.1.1 Home Page

This page will be the first page that all the users will see when they enter WICS. The administrator of WICS will own the home pages. This page will contain an introduction and welcome message to the users and provide links to various useful sites including the WICS. The page gives a brief overview of the system and its useability.

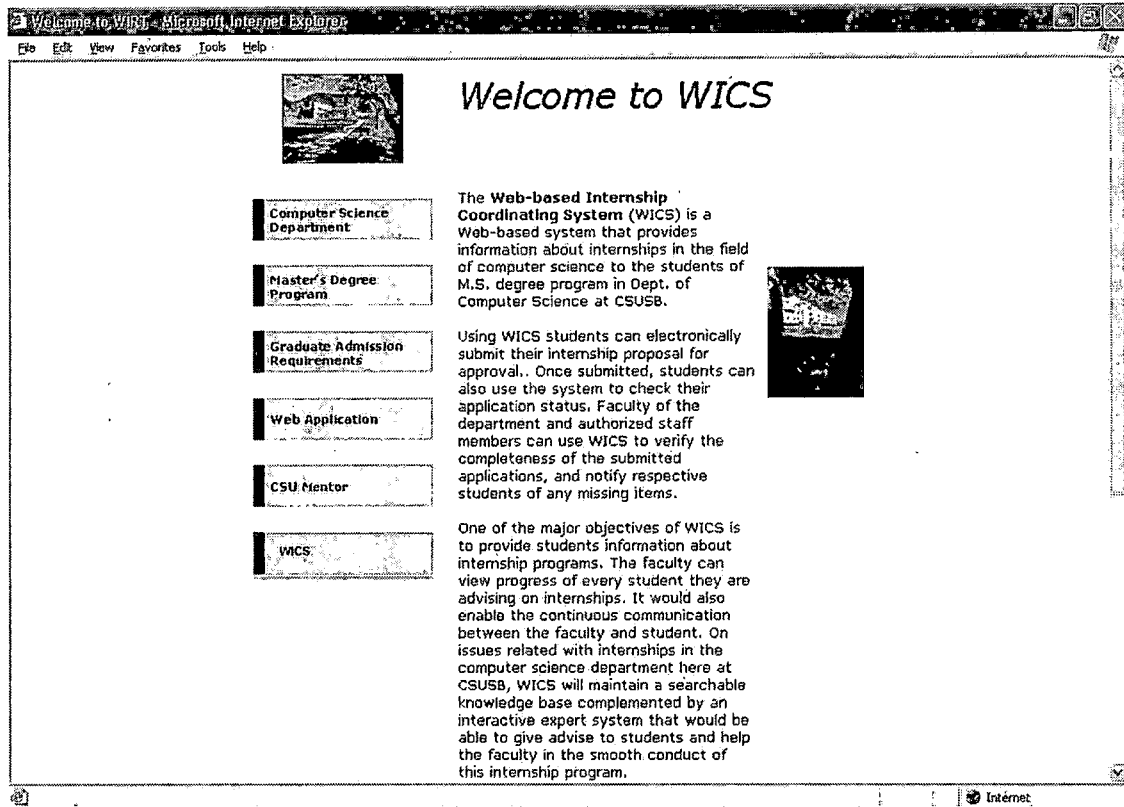


Figure 5. Home Page

4.1.2 Login Page

The user logs in by providing a user id and a password. After the authenticating system verifies the user id and password, it forwards to a PHP page, which will show the main page. Moreover, the user information will be saved in the session for later use, and the session will be killed when the browser is closed. The system will display different menu items based on the privileges granted to the user. If the user id or password is wrong, the program will show an error message and the

user can re-login. For guests, there is no need to check the database. The guest user who does not login can only browse the pages that are returned by the New Students link. This page also has other useful links for easy navigation. The link at the bottom left points to a help guide that can be updated regularly by the administrator.

Welcome to WICS - Microsoft Internet Explorer

File Edit View Favorites Tools Help

California State University, San Bernardino
Computer Science Department

Web-based Internship Registration Tool.

Existing Users:

Login:

Password:

Visiting Users:

New Students: [Click Here](#)

[HELP](#) | [CSUSB Home](#) | [CSCI Department Home](#) | [WICS Home](#)

Done Internet

Figure 6. Login Page

4.1.3 Administrator Login Page

When an administrator signs into the system, the authentication determines the type of user and accordingly redirects to the administrator login page. This page has

all the administrator privileges listed on the left. This page is dynamic as the name of the administrator is read from the database every time it is accessed.

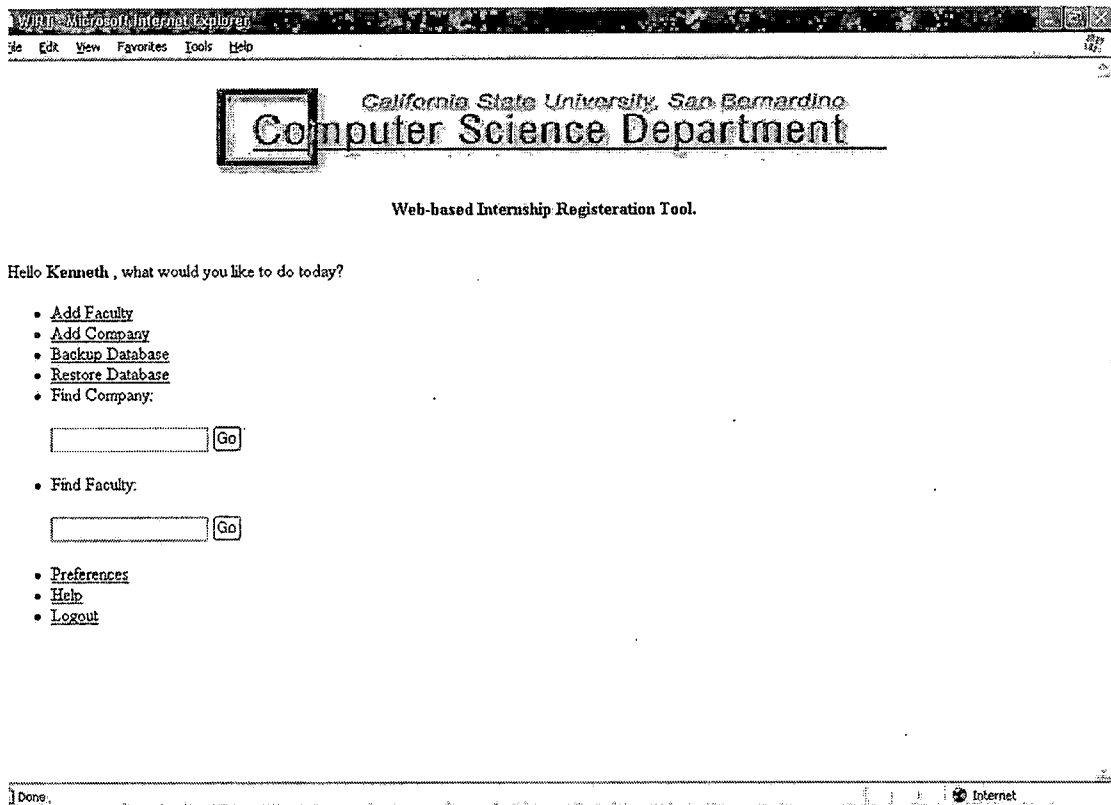


Figure 7. Administrator Login Page

4.1.4 Administrator Add Faculty Page

This page is visible only to the administrator and it provides an easy interface to add faculty information. New faculty information that is stored in the database includes name, email address and social security number.

WICS - Microsoft Internet Explorer

File Edit View Favorites Tools Help

California State University, San Bernardino
Computer Science Department

- [Home](#)
- [Add Faculty](#)
- [Add Company](#)
- [Backup Database](#)
- [Restore Database](#)
- [Find Company:](#)

Find Faculty:

- [Preferences](#)
- [Help](#)
- [Logout](#)

Enter New Faculty Information.

☐ - Required
☐ - Optional

☒ First Name:

☒ Last Name:

☒ Email:

☒ SSN:

Done Internet

Figure 8. Add Faculty Page

4.1.5 Backup Database Page

This is one of the features of WICS, which allows a user to create a backup of the entire database with a click of a button. This important feature maintains copies of the database and can be used again in case of a database failure.

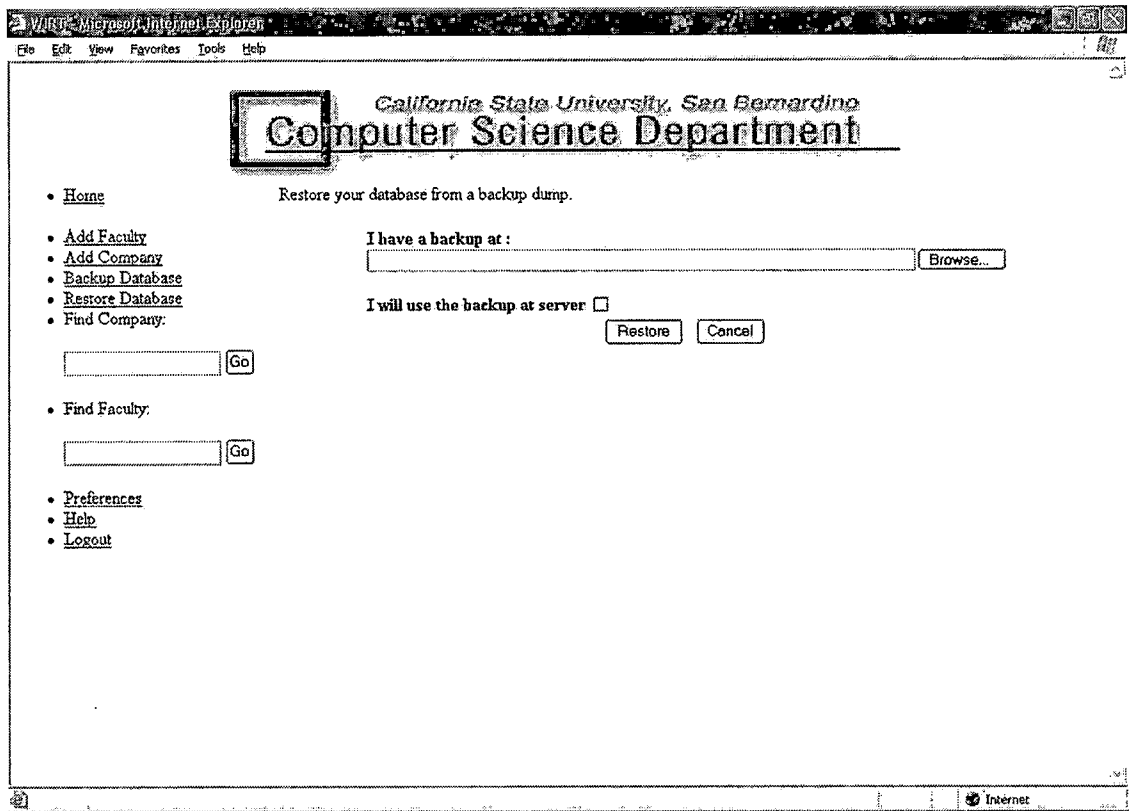


Figure 9. Backup Database Page

4.1.6 Faculty Search Result Page

The administrator can query the database for a complete search of all the faculty members. This page also allows the administrator to modify and delete faculty information. Care is taken that if a faculty is deleted then all the tables in the database having the faculty information are deleted in order to keep the database in a consistent state.

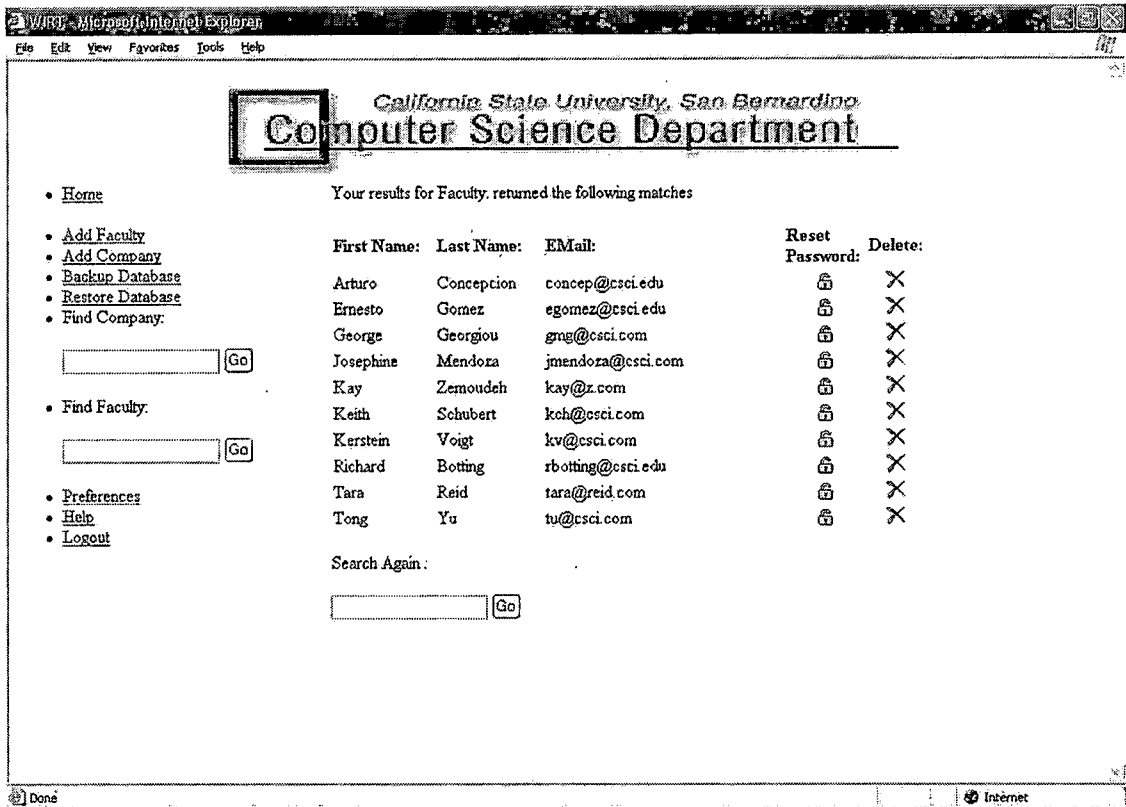


Figure 10. Faculty Search Result Page

4.1.7 Faculty Login Page

When the system identifies a user as a faculty member it gets redirected to the faculty login page. This page lists all the privileges assigned to a faculty member on the left side.

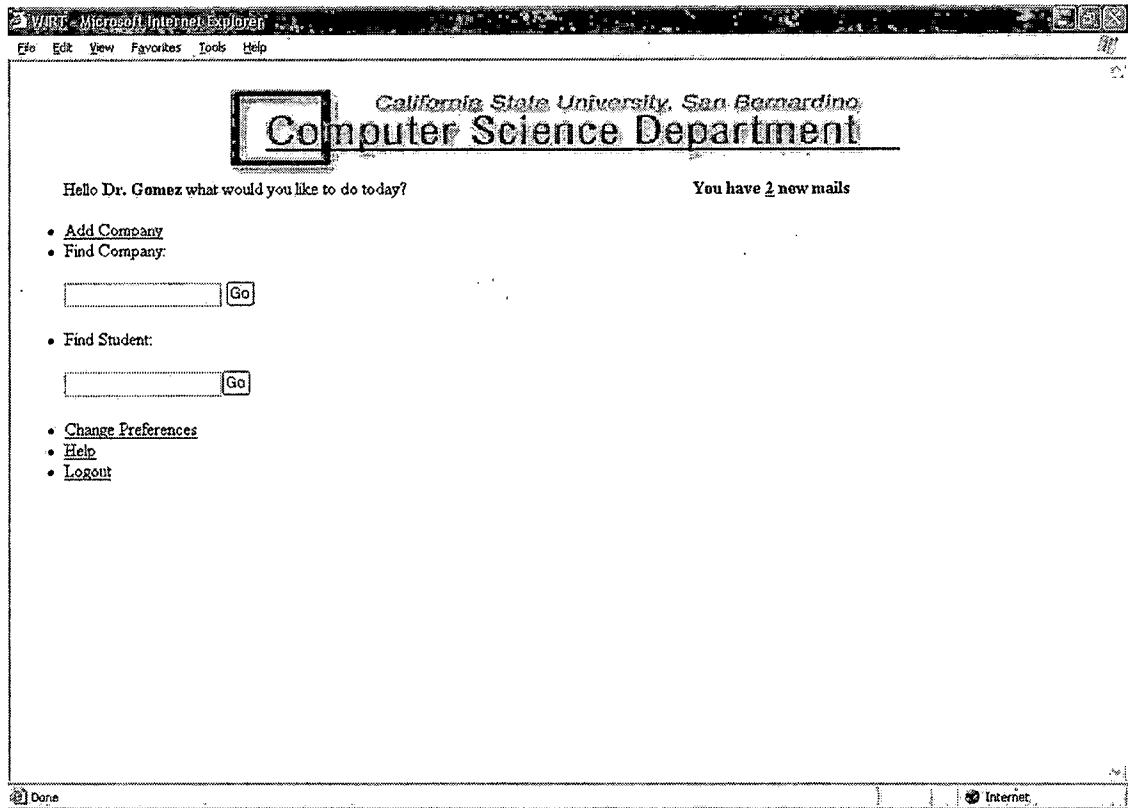


Figure 11. Faculty Login Page

4.1.1.8 Faculty View New Student Page

The faculty gets an alert whenever a new student submits a proposal for review. The faculty then has the option of either accepting or rejecting the proposal. Depending on the decision, the system automatically sends an email to the student with the necessary information.

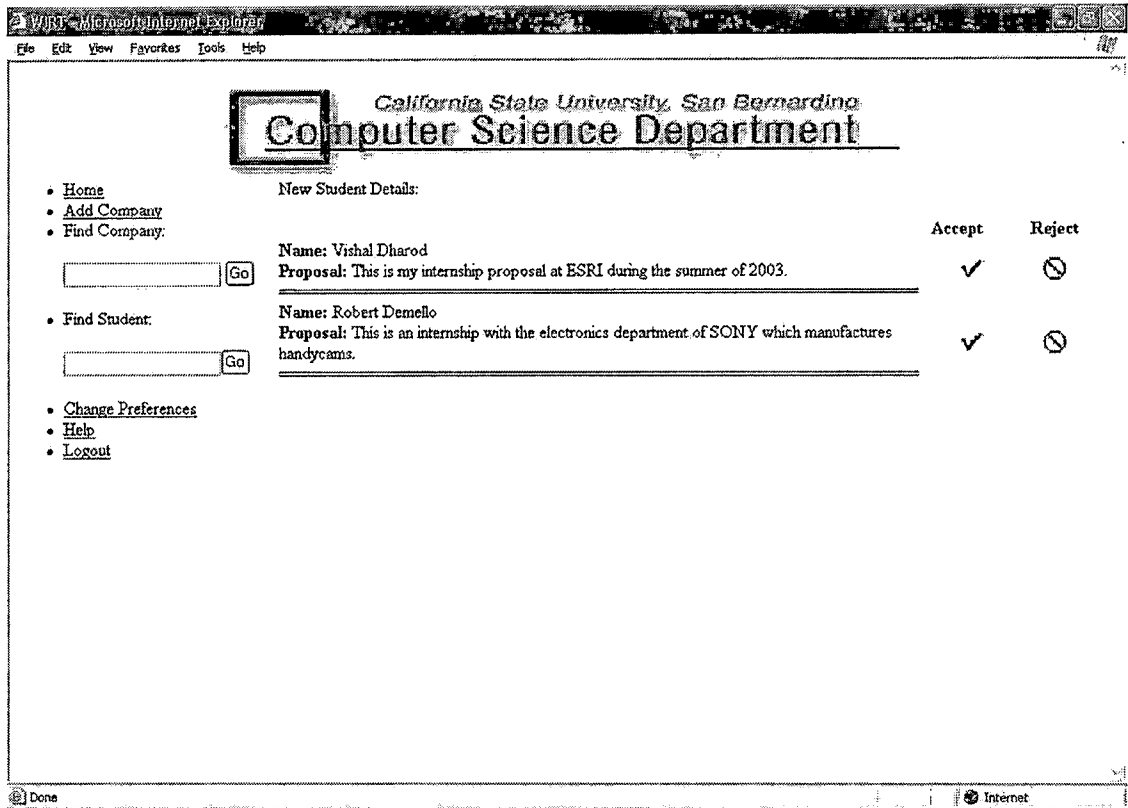


Figure 12. View New Student Page

4.1.9 Faculty View Student Progress Page

The View Student Progress page displays the progress made by a student in the internship class and further shows items that are still required by the student to successfully complete the class. The instructor can make changes to this page depending on the student's progress and send an email to the student if the need arises.

California State University, San Bernardino
Computer Science Department

- [Home](#)
- [Add Company](#)
- [Find Company:](#)

Student Progress Report for :

Name: Vishal Dharod **Student ID:** 666555999 [Send Email](#) [Delete Student](#)

- **Find Student:**

Company Name: Cisco Networks
Start Date: 02/5/2004
End Date: 04/5/2004

- [Change Preferences](#)
- [Help](#)
- [Logout](#)

<u>Documents:</u>	<u>Submitted:</u> <u>Yes / No</u>
Proposal	✓
Internship Offer Letter	✗
Application Form	✓
Letter from Field Supervisor	✗
Progress Letter	✗

Figure 13. View Student Progress Page

4.1.10 Student Search Result Page

The student search result page will only display a list of students the particular faculty member is advising. The name of the student alongwith the social security is displayed. This page provides a link to the view student progress page where the faculty can view and edit the student's progress.

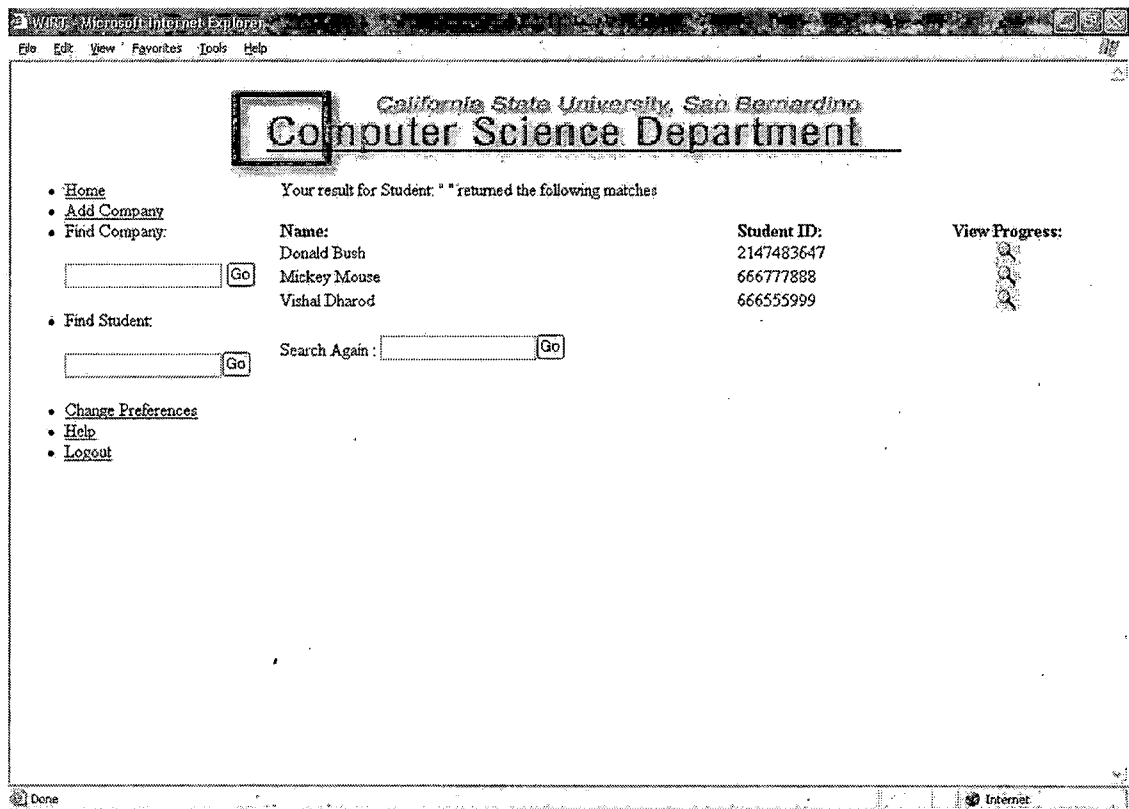


Figure 14. Student Search Result Page

4.1.11 Company Search Result Page

The administrator and faculty members have the privilege to add, modify, delete and view company information. The company search result page displays the entire list of companies that have been added to the database. The result is sorted in alphabetical order. Besides every company name, there is an option of modifying company information and deleting a company from the database if necessary.

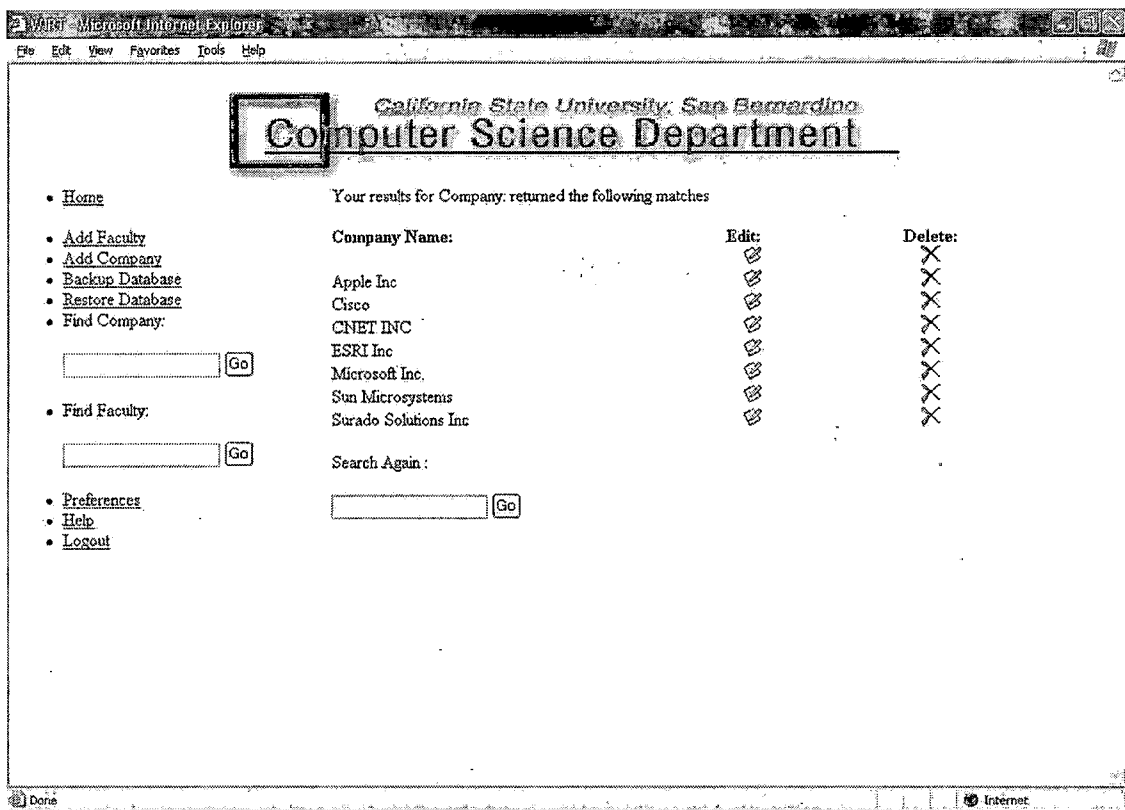


Figure 15. Company Search Result Page

4.1.12 Faculty Change Preferences Page

Faculty members can modify their personal information through this page. However, if the passwords entered do not match then appropriate error message is displayed and preferences are not changed. Error handling is of significant importance as this page updates sensitive personal information.

CMAC - Microsoft Internet Explorer

File Edit View Favorites Tools Help

California State University, San Bernardino
Computer Science Department

- [Home](#)
- [Add Faculty](#)
- [Add Company](#)
- [Backup Database](#)
- [Restore Database](#)
- [Find Company](#)
- [Find Faculty](#)
- [Preferences](#)
- [Help](#)
- [Logout](#)

Change your Preferences:

First Name: Kenneth

Middle Name: D

Last Name: Hanson

Password: ..

Password (verify): ..

Email: ken@csci.csusb.edu

Go

Go

OK Cancel

Done Internet

Figure 16. Change Preferences Page

4.1.13 Error Page

One of the features of WICS is in the way it handles errors in user information. WICS is equipped with a thorough server side error checking functionality. This feature helps in maintaining an error free database.

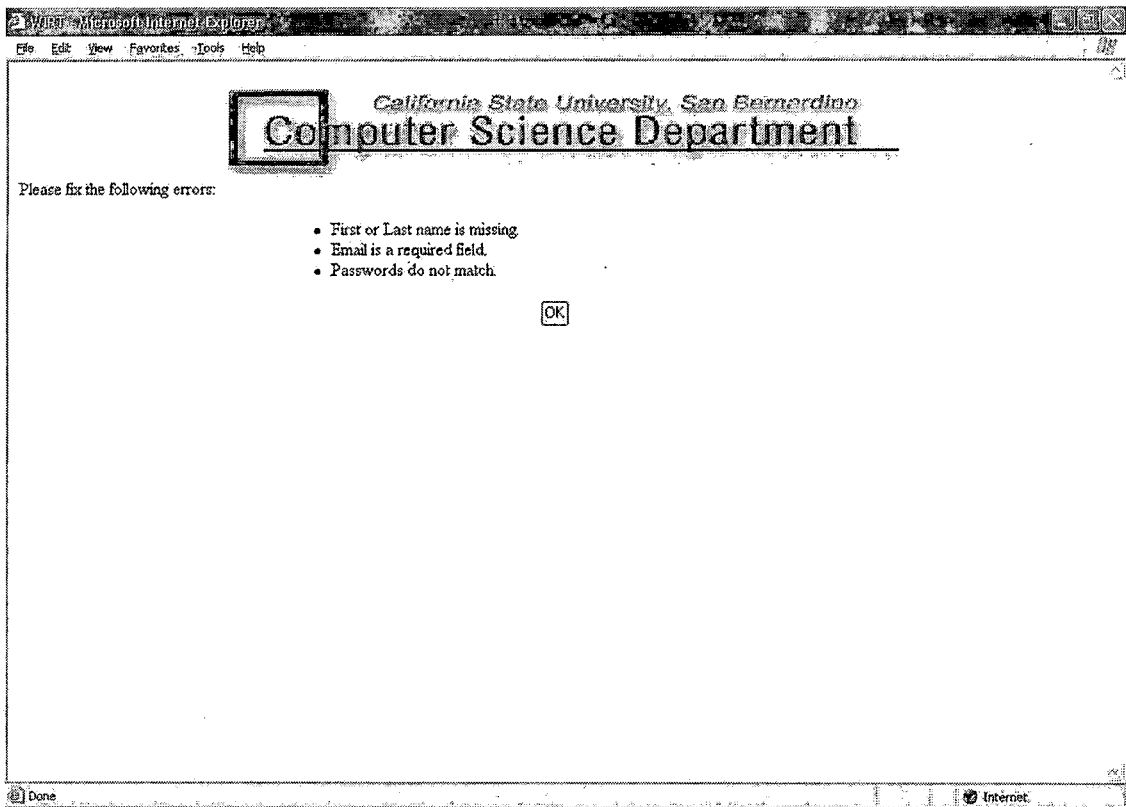


Figure 17. Error Page

4.1.14 Student Login Page

A student whose proposal is accepted by a faculty member is automatically added into the system. An email is sent out to the user with the login information. On successfully signing into the system, the student login page displays all the privileges a student has in this system.

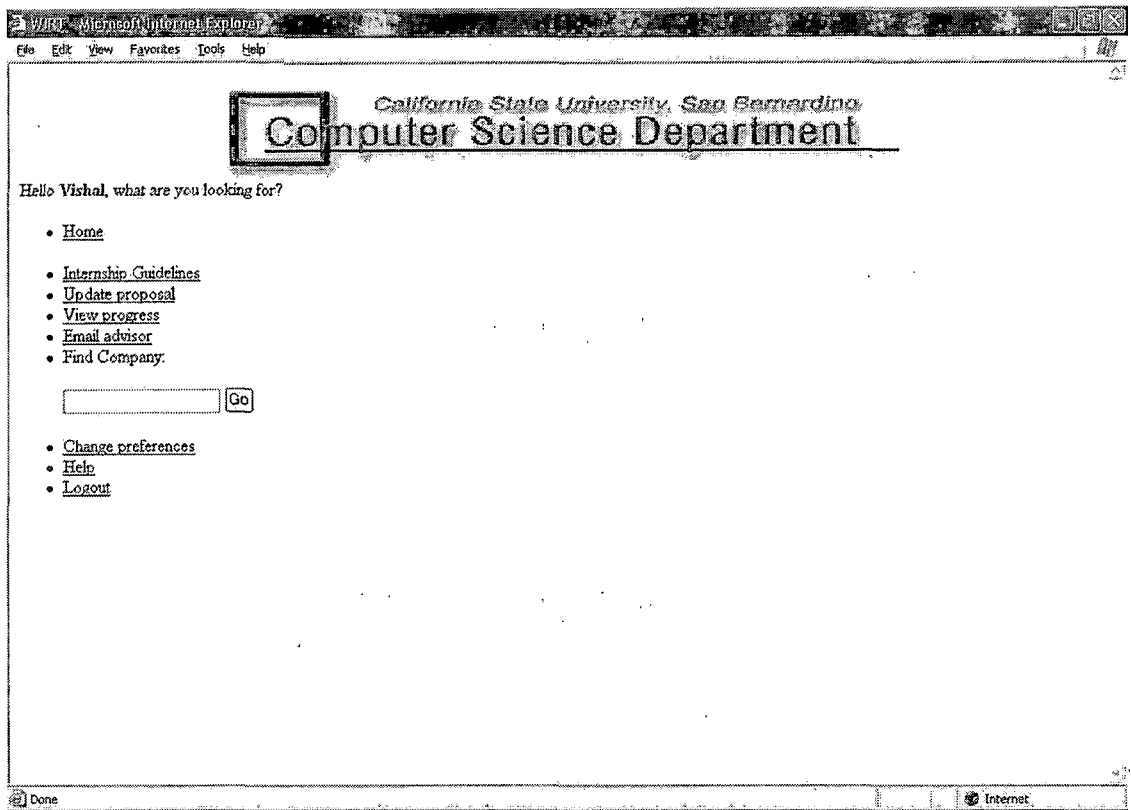


Figure 18. Student Login Page

4.1.15 Student Company Search Page

The company search result page is different for the students. The faculty and administrator of the system have the right to modify company information. However, students can only view the company information. The search result is sorted alphabetically and contains the name, address, phone number and a brief description of the company.

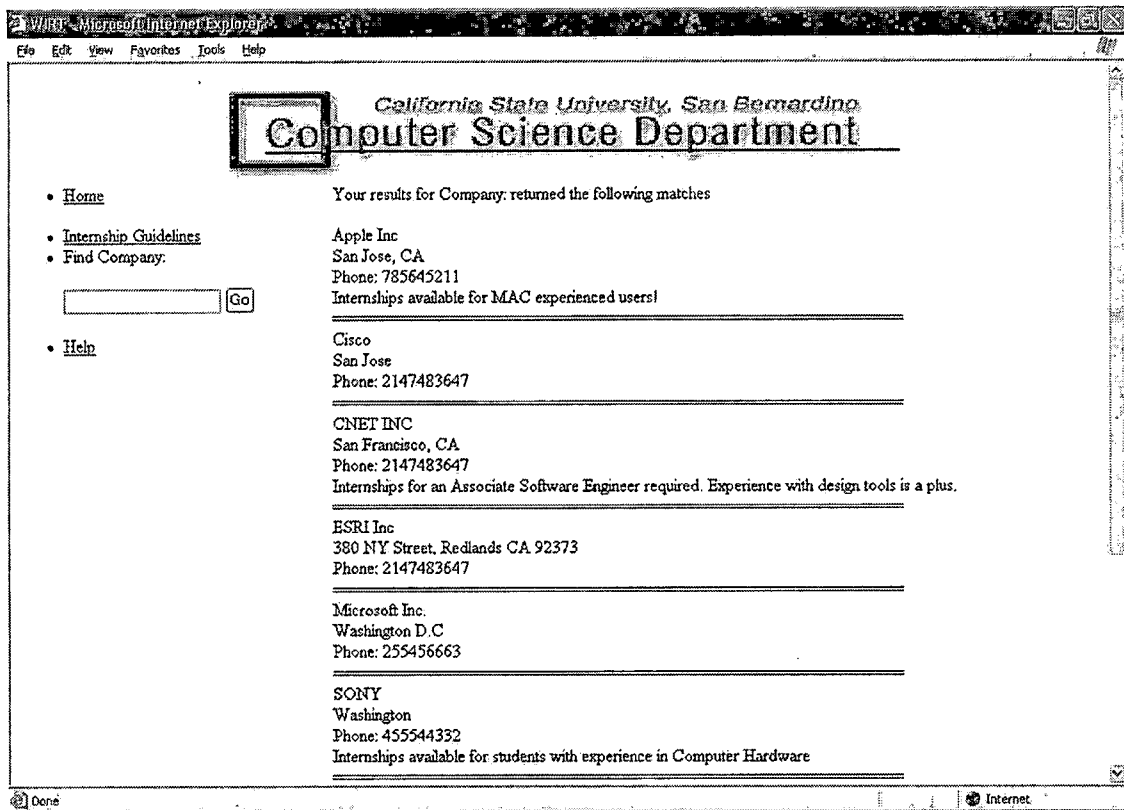


Figure 19. Student Company Search Result Page

4.1.16 Student View Progress Page

A student can view his/her progress in the internship class through this page. It is very similar to the page an instructor gets for a student's progress but the difference on this page for the student is that it is a read only page and no modifications can be done by the student. However, email facility is provided on this page for the student for any clarification to the concerned advisor.

California State University, San Bernardino
Computer Science Department

- [Home](#)
- [Internship Guidelines](#)
- [Update proposal](#)
- [View progress](#)
- [Email advisor](#)
- [Find Company:](#)

- [Change preferences](#)
- [Help](#)
- [Logout](#)

Your Progress Report :

Advisor Name:
Dr. Ernesto Gomez

Company Name: Cisco Networks

Start Date: 02/5/2004

End Date: 04/5/2004

[Send Email](#)

<u>Documents:</u>	<u>Submitted:</u> <u>Yes / No</u>
Proposal	✓
Internship Offer Letter	⊗
Application Form	✓
Letter from Field Supervisor	⊗
Progrss Letter	⊗

Enter Email Message Here

Figure 20. Student View Progress Page

4.1.17 Submit Proposal Page

A new student willing to register for an internship class can propose the idea to a faculty member through this page. The faculty is alerted upon receiving a request and has the right to either accept or reject the student's proposal. If accepted, then the student is added to the database by the system.

Microsoft Internet Explorer

File Edit View Favorites Tools Help

California State University, San Bernardino
Computer Science Department

• [Home](#)

• [Internship Guidelines](#)

• [Find Company:](#)

• [Help](#)

Submit an Internship Proposal.

☐ - Required
☐ - Optional

First Name:

Last Name:

SSN:

Email:

Phone No:

Advisor:

Company Name:

Proposal:

Done Internet

Figure 21. Submit Proposal Page

4.1.18 Proposal Submitted Page

The submit proposal page accepts a tentative internship proposal and submits it to the concerned faculty member. The student receives a message from the system saying that the proposal will be evaluated and a notification will be sent out soon after that.

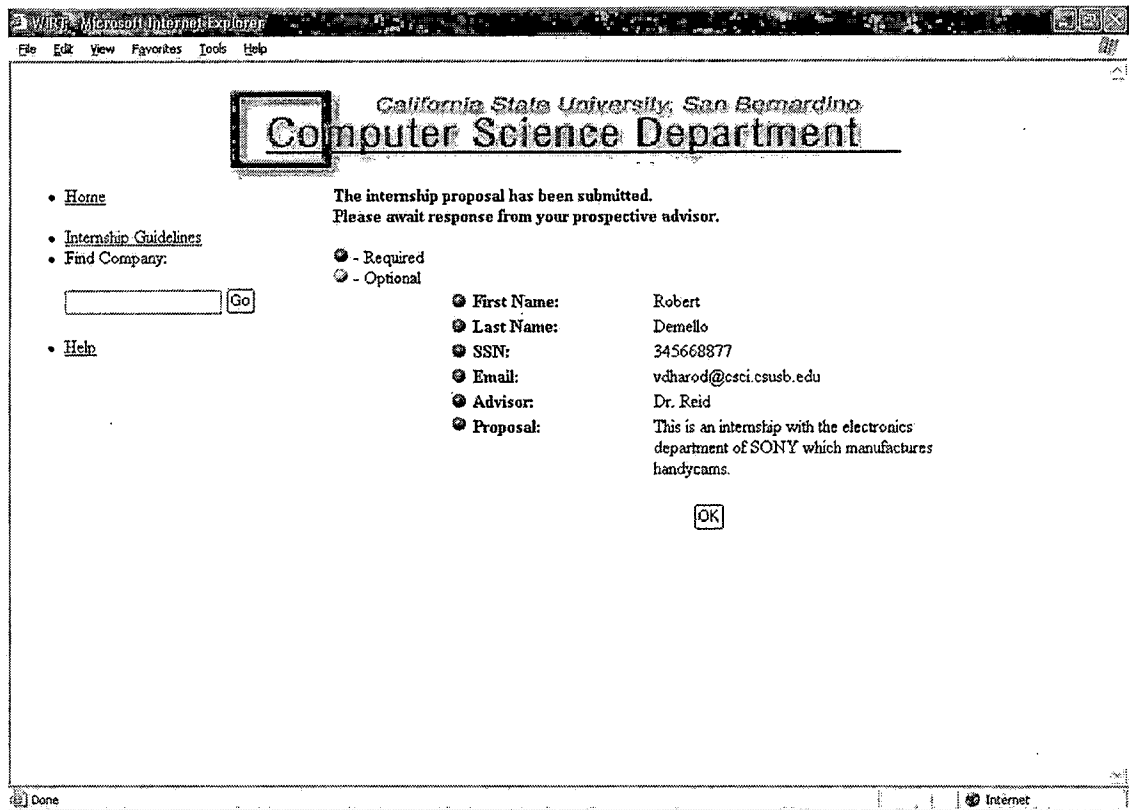


Figure 22. Proposal Submitted Page

4.1.19 Student Error Page

Any user input error should be caught and reported back to the user. The system has a solid server side error handling functions and any error encountered during data input is reported back to the student.

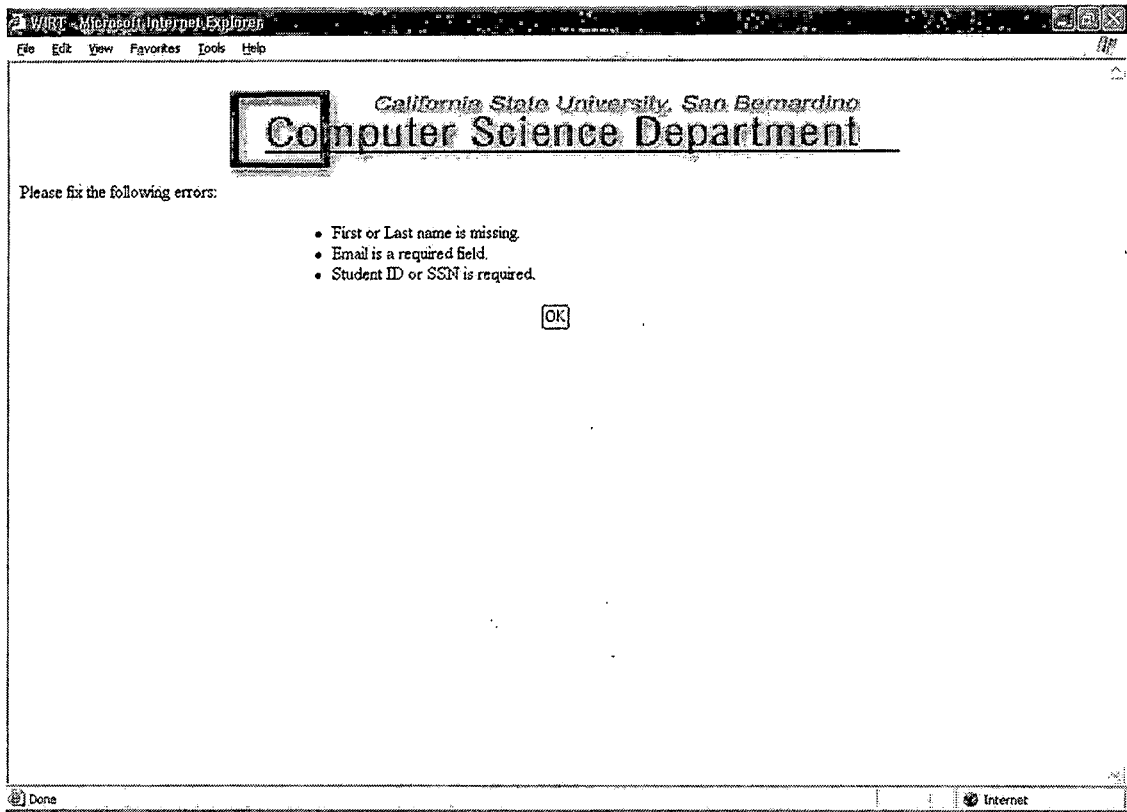


Figure 23. Student Error Page

4.1.20 Internship Guidelines Page

This page contains the most important information not only for the new students but also for the existing students who are registered for the internship class. The administrator of WICS has the ability to modify this page from time to time. It contains information on successfully completing an internship class.

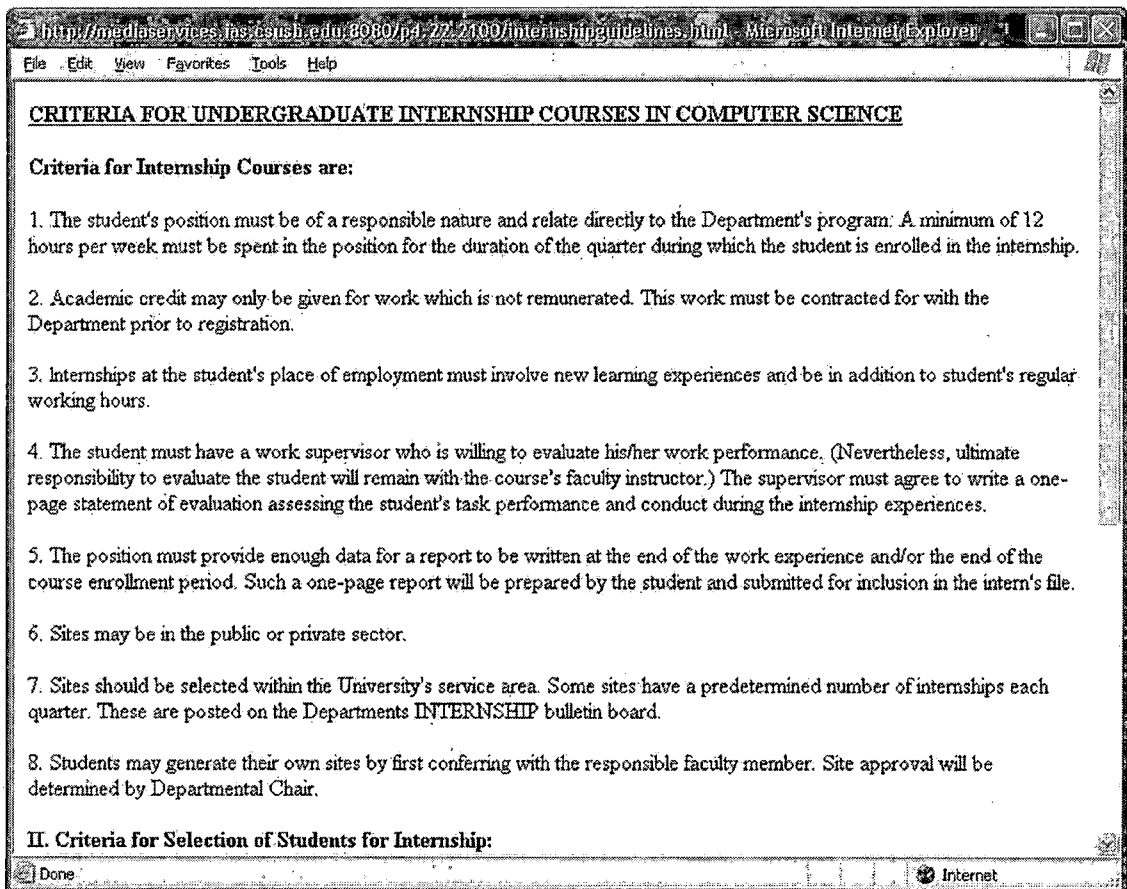


Figure 24. Internship Guidelines Page

4.1.21 Help Guide

This page is divided into three different categories. The first section contains useful points for the administrator. The second section is dedicated to the faculty members and it lists some of the frequently asked questions. The students can browse through the third section where some tips on using the system are listed.

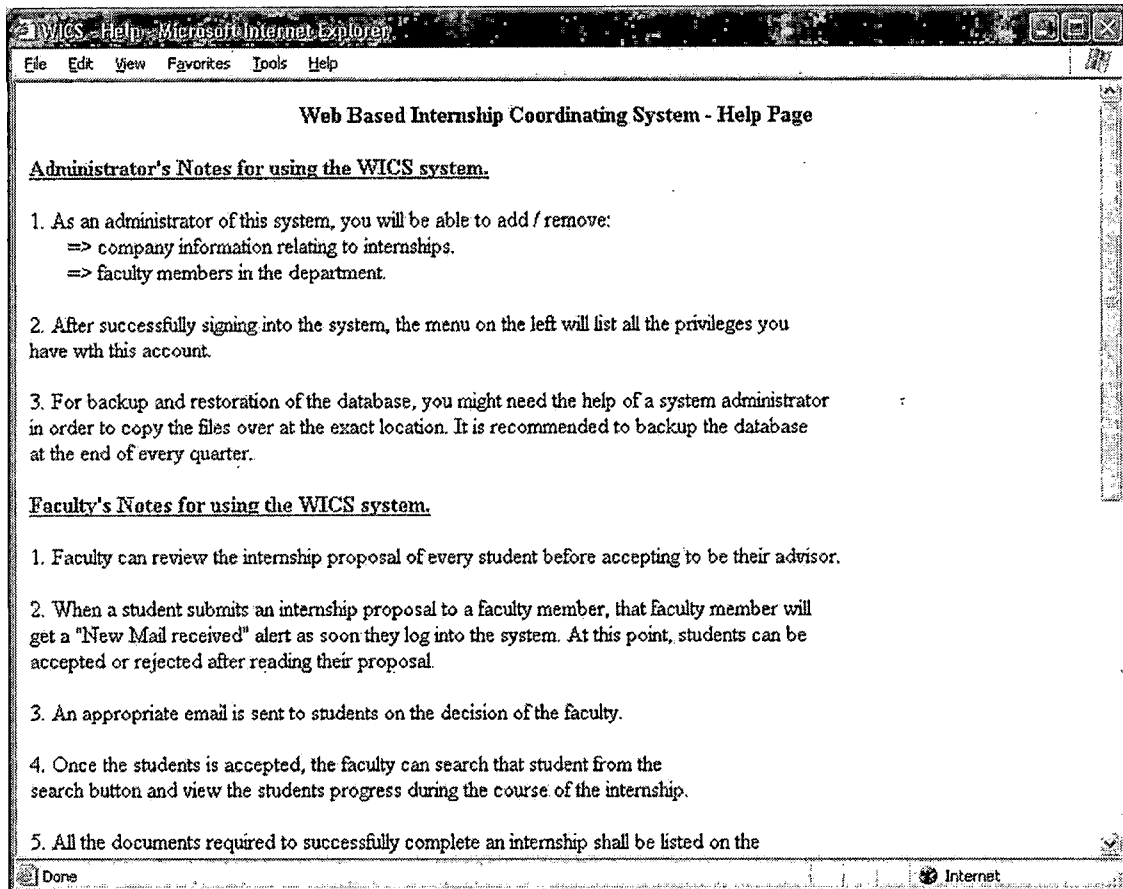


Figure 25. Help Guide

CHAPTER FIVE

SYSTEM VALIDATION

The system validation test is a kind of test process that ensures that the software program meets the expectation of the user. The purpose of system validation is to provide the highest degree of assurance that a specific process will produce the same result consistently and meet predetermined specifications and quality attributes. This can also guarantee the system performance and reliability. This validation process helps in eliminating lot of bugs from the code base and tries to make the software bug free. Meeting security standards is not possible without having a validation process and protecting sensitive user data from malicious users is the primary goal of testing.

5.1 Unit Test

Unit test is the basic level of testing where individual components are tested to ensure that they operate correctly. These individual components can be object, class, program, etc. The unit testing results of WICS are shown in Table 23.

Table 7. Unit Test Results (Forms)

Forms	Tests Performed	Results
Add Faculty Page	<ul style="list-style-type: none"> • Verify handling valid data input. • Check all the buttons work properly. • Check for errors on the error page. 	Pass
Title Page	<ul style="list-style-type: none"> • Check all the menus are shown properly to the user. • Check all the links work as expected. 	Pass
User Login Page	<ul style="list-style-type: none"> • Verify that appropriate page is loaded depending the role of the user. • Check user privilege before loading page. • Redirect to an error page if login information is incorrect. 	Pass
Edit Company Page	<ul style="list-style-type: none"> • Check the insert function enters correct data in the database. • Check all the button work properly. 	Pass
Session Page	<ul style="list-style-type: none"> • Check all the button work properly. • Verify the page can get the error message and work properly by the message. • Verify the user save in session after login • Verify logout button on all pages 	Pass
Logout Page	<ul style="list-style-type: none"> • Check if the session is killed and the user redirected to the default login page. 	Pass

Forms	Tests Performed	Results
Administrator Account Page	<ul style="list-style-type: none"> • Verify showing the correct administrator account information. • Verify the backup database link for correct implementation. 	Pass
Change Preferences Page	<ul style="list-style-type: none"> • Check all the buttons work properly. • Verify the page get the correct user account information. • Verify handling valid data input. • Verify the data updated correctly. 	Pass
Submit Proposal Page	<ul style="list-style-type: none"> • Verify all the required information is entered by the student. • Check if the link of the orphan page directs to correct page. 	Pass
Proposal Submitted Page	<ul style="list-style-type: none"> • Confirm that the new student cannot view any pages of a registered user. • Verify error checking has been thoroughly completed. • Make sure the user gets a confirmation notice. 	Pass
Faculty View New Student Page	<ul style="list-style-type: none"> • Verify the alert notice appears every time a new proposal is submitted. • Make sure the buttons work as expected 	Pass
Faculty Decision Page	<ul style="list-style-type: none"> • Make sure the faculty member gets alerted upon receiving any new proposals. • Check for alternative messages when mouse is pointed to any of the buttons. • Make sure all the buttons work correctly. 	Pass

Forms	Tests Performed	Results
Email Delivery Notice	<ul style="list-style-type: none"> • Make sure the student gets an appropriate email based on the faculty's decision. • Check to see whether the correct email address is being used. • Make sure all the buttons work correctly. 	Pass
Add Company Page	<ul style="list-style-type: none"> • Make sure error is reported for any missing fields that are required. • Make sure all the buttons work correctly. 	Pass

Table 8. Unit Test Results (Class: DataBase)

Functions	Tests Performed	Results
Database Connection	<ul style="list-style-type: none"> • Check to see if the database is running. • Verify that it can handle several requests concurrently. 	Pass
Data type error	<ul style="list-style-type: none"> • Verify fields get the appropriate data as their datatype. • Throw an error if the data is not consistent with the data type. 	Pass
Data Availability	<ul style="list-style-type: none"> • Check for availability of data at all times. • Make data available only after authorizing access. 	Pass

Functions	Tests Performed	Results
Delete Company	<ul style="list-style-type: none"> • Check if the company is deleted from all the tables. • Make sure it can handle the request of deleting a non-existent company. 	Pass
Delete Faculty	<ul style="list-style-type: none"> • Check if the user is deleted from all tables. • Make sure the deletion takes place without any referential integrity constraint issues. • Make sure all students under that faculty member are deleted as well. 	Pass
Update User Info	<ul style="list-style-type: none"> • Check if the data is updated. • Verify if the updated value is correct. 	Pass

5.2 Subsystem Testing

Subsystem testing is the next step up in the testing process where all related units from a subsystem do a certain task. Thus, the subsystem test process is useful for detecting interface errors from a front-end perspective and specific functions from the back end point of view. Table 9 shows subsystem test results in detail.

Table 9. Subsystem Test Results

Subsystem	Tests Performed	Results
Authorize subsystem	<ul style="list-style-type: none"> • Test if error message is displayed on incorrect login. • Make sure the result of authorizing user is correct. • Verify the login user information is store in session properly. • Check if proper role is assigned to the user upon successful login. • Verify the login page redirect to the correct browsing page after the user logs in. 	Pass
Accounts management subsystem	<ul style="list-style-type: none"> • Make sure all the existing users are listed in the user list. • Check if the subsystem can detect the error of creating of the user that exists in the subsystem. • Check if the user can update his/her own account properly. • Verify the newly created user information is the same as the information provided. 	Pass
Browsing subsystem	<ul style="list-style-type: none"> • Check if the subsystem checks for user privilege before showing pages. • Verify the page is showing properly after the user click on the page link. 	Pass
Editing subsystem	<ul style="list-style-type: none"> • Make sure the subsystem checks the user privilege before forwarding to edit page. • Verify the subsystem check the user privilege before update the page information. • Verify if the subsystem shows the page properties. 	Pass

5.3 System Testing

System testing is the testing process that uses real data, which the system is intended to manipulate, to test the system. First all subsystem will be integrated into one system. Then test the system by using a variety of data to see the overall result. This testing is usually done when the system is about to be deployed in a production environment. The deployment is a mere replication of the tested system into production servers but tests on production environment have to be done to make sure everything is running.

System testing of WICS system begins with the following steps:

Table 10. System Test Results

System Testing		Results
1.	Install WICS system into server.	Pass
2.	Start up all servers such as Apache Web server, Mysql database server.	Pass
3.	Run tests by using real data on all forms and reports.	Pass

CHAPTER SIX

MAINTENANCE MANUAL

It is very important to have a maintenance manual with a system regardless of how easy to use the system is. The maintenance manual records all the information that is used to setup or backup the system. In order to make sure the system works smoothly, meets the expectation of the users and maintains sensitive data securely, it is extremely important to follow the maintenance manual step by step carefully. In WICS, there are 4 major issues: Hardware Selection, Software Installation, Variable Installation, and WICS Installation.

6.1 Software Installation

To deploy WICS, it is necessary to know the amount of load the system will have on any given day. Based on this proper hardware should be dedicated to run WICS on. WICS requires RedHat 9.0 or higher, Mysql, PHP, PhpMyAdmin, APACHE, and ODBC to run the programs. Following will detail the installation of those 6 softwares.

6.1.1 RedHat 9.0 Installation

RedHat is a linux base operating system that is offerd freely and be downloaded from Internet. The reason we choose RedHat is because it is an open source and so

does not have to be purchased. Moreover security issues can be better addressed in Unix based Operating Systems.

Following are the steps to install RedHat:

1. Download a latest version of the RedHat 9.0 operating system from <http://ftp.redhat.com/pub/redhat/linux/9/en/iso/i386/> and burn all the files into 3 CD-Roms.
2. Install the operating system by inserting CD 1 into the CD-ROM drive and boot (start up) the machine (server) that is going to install the operating system.
3. The machine shall startup via CD-ROM and begins installing RedHat 9.0.
4. Follow the install wizard and enter the required information such as network setting and the hardware environment.
5. After all the necessary files are copied into the computer and install completes, the machine will restart and Redhat installation will end.

6.1.2 Database Installation

Mysql is the database system that is used in WICS; it's free, and is included in all the versions of RedHat by default. The reason that we choose Mysql as WICS's database system is because it also provides a ODBC driver

up the database system. Login wics as the supervisor and create a database user, "wics," and the database "wics."

3. There are still some steps needed to setup the environment values.

In the user's environment setup file /etc/profile.d/*.sh, add the following line:

```
export PGDATA=/var/lib/mysql/data
```

Open the file /var/lib/mysql/data/Mysql.conf and uncomment the line:

```
tcpip_socket = true
```

In order to have the database system starup at the system start, have the command executed:

```
/sbin/chkconfig --level 3 Mysql on
```

And, the last step is to startup the database system immediately without restarting the system:

```
/sbin/service mysqld start
```

After having the steps above executed, the database system is ready to go and now we have to install PHP, Hypertext Preprocessor. (A HTML embedded scripting language)

6.1.3 Hypertext Preprocessor Installation

PHP is a HTML embedded scripting language and it runs on the APACHE Web Server. PHP just like Mysql comes along with the Red Hat CD Rom, hence selecting PHP packages while installing Red Hat would be a very good idea.

However, if PHP is not installed then go to <http://php.net> to download the latest version of PHP for Linux (all languages, including English) to the directory /usr/bin, then execute the following commands:

```
rpm -ivh php_version-linux-i586.rpm
```

6.1.4 Apache Web Server

APACHE is one of the apache jakarta projects, which is a web container to process PHP programs and to serve static web pages. First of all, we go to the Apache's official download ftp server at <http://ftp.epix.net/apache/Apache-5/v5.0.12-beta/bin/> to download the file of Apache server for linux Apache-5.0.12.tar.gz to /usr/bin/ and extract it to the hard drive.

```
tar -xzf Apache-5.0.12.tar.gz
```

Add the following lines in the file /etc/rc.local to have the Apache run when the system boots:

```
export APACHE_HOME=/usr/bin/Apache-4.1.27  
${APACHE_HOME}/bin/startup.sh
```


6.1.5 Open Database Connectivity

The API used to execute SQL statement is different for each database engine. PHP programmers, however, are lucky and are freed from such database portability issues. They have a single API, the Open Database Connectivity API (ODBC), that's portable between database engines. The ODBC library provides an interface for executing SQL statements. It provides the basic functionality for data access. A number of drivers are available for Mysql, and information about this can be obtained at the Mysql homepage at <http://ODBC.mysql.org/download/>. However, ODBC drivers are included in the Red Hat CD Roms. By selecting the correct ODBC driver during installation, you are saved from the hassle of installing and configuring the driver manually after installation. If installed directly from the Red Hat package, the configuration files are automatically modified to connect to PHP programs.

6.2 Variables Modification

In WICS, we have to change some environment variables in the linux system and httpd.conf file in Apache server configuration directory.

6.2.1 System Variables

1. Open the file "httpd.conf" in the directory
"/etc/httpd/conf" using "vi" or any other text editor.
2. Scroll down until you see the document root and change it to the directory path where WICS is going to be installed.
3. The variable "ServerName" indicates the name the server will be referred to. Change that to the name you decide for the server.
4. The variable "Listen" refers to the ip (internet protocol) address and port number the server will listen for requests on. Change the default setting to the ip address of your machine and assign a port number to it.

6.3 System Installation/Migration

1. All the PHP programs and HTML programs are stored in the directory
var/www/html

6.4 Backup and Restore

Backup is a very important functionality needed for any system to prevent losing important and sensitive data. This slightly complicated feature is made very easy in

WICS. By the click of a button, the entire data is backed up in a SQL file and can be opened in any text editor. The other step is to backup the database by entering the backup command at prompt.

6.4.1 System Backup

All the WICS system files are located in the directory `var\www\html` and all its subdirectory. Thus, in order to backup the system files, all we need to do is to backup the files in the directory. The method suggested is to compress the directory `"var\www\html"` including its subdirectory to compress files for future use by the compress program `"tar."` Using the following command to backup the system files:

```
tar -cf WICS.tar \var\www\html
```

6.4.2 Database Backup

To backup the database system, we use `mysqldump` command to backup the database used by WICS. The following command is used to backup the database:

```
mysqldump wics -uroot -ppassword | gzip > WICS.zip
```

After executing the backup command above, the file `WICS.zip` would be the backup file of the database.

6.4.3 System Restore

To restore the system file, simply extract the backup file by using the following command:

```
tar -xzvf WICS.tar /
```

By the command above, all the WICS system files will restore into the directory `\var\www\html` and complete the restore system process.

6.4.4 Database Restore

To restore the database needed for the WICS, go to the directory where your database backup file is in, and execute the following commands:

```
Create database wics
```

```
gunzip -c WICS.zip | mysql WICS
```

After the commands are executed, the database is restored to the database system. Then, Apache web server needs to be restarted and WICS will be completely restored.

CHAPTER SEVEN

CONCLUSION AND FUTURE DIRECTIONS

7.1 Conclusion

WICS provides a very good communication environment for the instructor and the students who wish to register for an internship class. For the instructor, WICS offers a very good environment to monitor a student's progress and avoid unnecessary delays. For the students, WICS provides a good environment that all the students can look at the shared information about companies. Students can read the internship guidelines online and understand all that it takes to successfully complete an internship. The system is very easy to use and any additional requirements can be easily added without affecting the current information. This is what makes it so convenient to use. The administrator of the system has the privilege to backup the entire system with a click of a button. Only a faculty member has the privilege to modify a student's progress and at every step, the automated email function sends out emails to students, faculty and administrator of any changes pertaining to them. The session checked function on every page ensures that the user is authorized to view the contents of that page and thus prevents any malicious

activity with sensitive user data. WICS is definitely a good tool that will help the instructor and the students have a better communication environment.

7.2 Future Directions

WICS is a system that can be used by any department offering an internship class. In the future, a progress report can be made available to the student in Portable Document Format (PDF). This will include a comprehensive summary of the student's progress and mention tasks that are remaining as well. The view progress page for both the faculty and students can be made more graphical and a bar graph or a pie chart can be displayed to show the progress and have the user better interpret the result. WICS can also have an interface for adding more requirement documents into the internship class and assign the administrator the privilege of adding such documents. WICS should also make a note of the time period of the internship for the faculty to judge the number of units it could be worth for.

After testing this system out in one department, it can be deployed by the entire university and make it mandatory for students to use the system for an internship class. Before this happens though, the requirements of

hardware and security issues need to be reviewed for the system to run successfully. It would also be strongly recommended to have a person maintain this system if deployed in a big production environment. This could be the administrator of the system.

APPENDIX A
USER CLASS PRINTOUT


```

<?php
class User {

    var $username;
    var $firstname;
    var $role_id;

    function addUser($uname,$fname,$role) {
        $this->username = $uname;
        $this->firstname = $fname;
        $this->role_id = $role;

    } // function addUser

    function getFN() {
        return $this->firstname;
    } // getFN

    function getUN() {
        return $this->username;
    } // getUN

    function getRole() {
        return $this->role_id;
    } // getRole

} // class user
?>

```

APPENDIX B
DATABASE CLASS PRINTOUT

```

<?php
class database
{
    var $host;
    var $user;
    var $passwd;
    var $dbname;
    var $link;
    var $result;

    function database($host, $user, $passwd, $dbname)
    {
        $this->host = $host;
        $this->user = $user;
        $this->passwd = $passwd;
        $this->dbname = $dbname;
    }

    function connect()
    {
        $this->link = mysql_connect($this->host, $this->user, $this->passwd) or
            die("Could not connect to database.");
        mysql_select_db($this->dbname) or
            die("Could not open database.");

        return $this->link;
    }

    function execute($sql)
    {
        if($this->result = mysql_query($sql, $this->link))
            return $this->result;
        else
            return false;
    }
}
?>

```

REFERENCES

1. Martin Fowler with Kendall Scott. UML Distilled - A brief guide to the standard object modeling language. Addison Wesley Longman, July 2001.
2. Laura Thompson. PHP and MySQL Web Development. SAMS Publishing, February 2003.
3. Elmasri and navathe. Fundamentals of Database Systems, third edition. Addison Wesley, June 2000.
4. PHP Reference Manual.
<<http://www.php.net/>>
5. William B. Sanders. JavaScript DESIGN. New Riders, 2002.
6. Mysql Reference Manual for version 7.3.
<<http://www.Mysql.com/> >